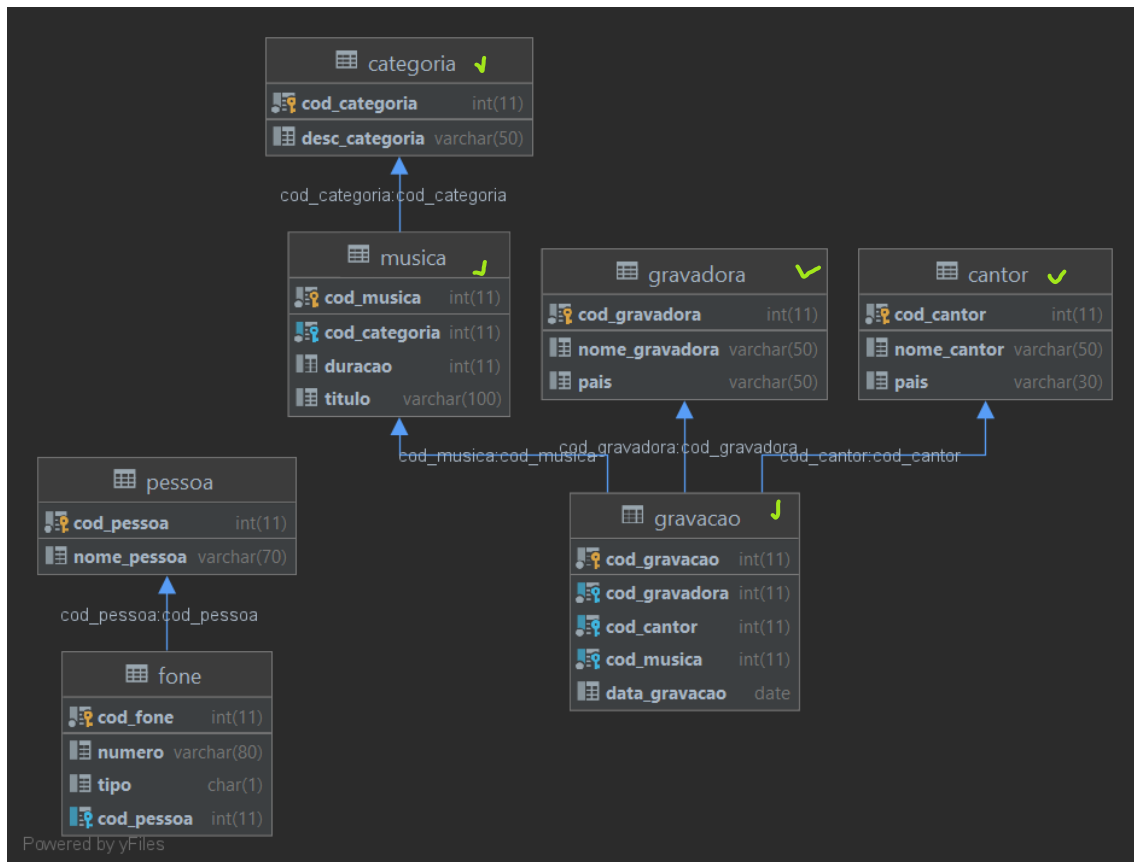


**ESTE DOCUMENTO POSSUI 2 PÁGINAS, LER NA ÍNTEGRA ANTES DE INICIAR A ATIVIDADE
(NÃO ENTREGAR ESSE SIMULADO)**



Utilizando o cenário da Atividade 1 e de acordo com o Diagrama Entidade-Relacionamento acima, Faça o seguinte:

1. Crie as classes com as respectivas anotações em JPA a partir do DER apresentado acima para as Entidades (`@Entity`); e para os Relacionamentos que existem entre elas (**obrigatoriamente criar o relacionamento bidirecional - `mappedBy`**).
2. Crie as interfaces de repositório (`@Repository`) para as classes de modo a permitir o acesso aos métodos de CRUD do `JpaRepository`.
3. Crie as classes de serviço para cada classe Entidade que foi implementada na questão 01 e que irá formar a camada de lógica de negócios (`@Service`). Utilize na implementação de cada classe a injeção de dependências com a camada de acesso a dados implementada (`@Repository`) por meio da declaração de um atributo usando a anotação `@Autowired`.
4. Crie em cada classe da camada da lógica de negócios (`@Service`), um método para salvar a entidade.
5. Crie em cada classe na camada de lógica de negócios (`@Service`), um método para excluir a partir do identificador - delete(ID) - um registro de cada entidade, utilizando injeção de dependências com a camada de acesso aos dados (`@Repository`).

6. Crie em cada classe na camada de lógica de negócios (@Service), pelo menos 2 (dois) métodos com consultas para cada entidade, utilizando injeção de dependências com a camada de acesso aos dados (@Repository). Quando necessário, acrescentar a assinatura dos métodos na camada de acesso aos dados.
7. Usando o CommandLineRunner e o Logger, realize operações de inclusão, exclusão e consulta para todos os métodos criados em cada classe da camada de lógica de negócio criada nas questões anteriores.

IMPORTANTE:

- Para realizar essa atividade, recomendamos ler a apostila “Implementando relacionamentos bidirecionais com JPA” que está disponibilizado essa semana.